Intruder deducibility constraints with negation. Decidability and application to secured service compositions.

Tigran Avanesov , Yannick Chevalier , Michaël Rusinowitch ,
Mathieu Turuani

ISSN 0249-6399   ISRN INRIA/RR--8017--FR+ENG

RESEARCH
REPORT
N° 8017
July 2012
Project-Teams CASSIS

Intruder deducibility constraints with negation. Decidability and application to secured service compositions.*

Tigran Avanesov † ‡ §, Yannick Chevalier ‡, Michaël Rusinowitch †, Mathieu Turuani †

Project-Teams CASSIS

Research Report n° 8017 — July 2012 — 21 pages

Abstract: The problem of finding a mediator to compose secured services has been reduced in our former work to the problem of solving deducibility constraints similar to those employed for cryptographic protocol analysis. We extend in this paper the mediator synthesis procedure by a construction for expressing that some data is not accessible to the mediator. Then we give a decision procedure for verifying that a mediator satisfying this non-disclosure policy can be effectively synthesized. This procedure has been implemented in CL-AtSe, our protocol analysis tool. The procedure extends constraint solving for cryptographic protocol analysis in a significative way as it is able to handle negative deducibility constraints without restriction. In particular it applies to all subterm convergent theories and therefore covers several interesting theories in formal security analysis including encryption, hashing, signature and pairing.

Key-words: Web services, orchestration, security policy, separation of duty, deducibility constraints, cryptographic protocols, formal methods, tool

* This work is supported by FP7 AVANTSSAR [4] and FP7 NESSoS [20] projects.

† INRIA Nancy Grand Est, France. Email: {rusi, turuani}@inria.fr

‡ IRIT, Université de Toulouse, France. Email: ychevali@irit.fr

§ SnT, Université du Luxembourg, Luxembourg. Email: tigran.avanesov@uni.lu

RESEARCH CENTRE NANCY – GRAND EST

615 rue du Jardin Botanique
CS20101
54603 Villers-lès-Nancy Cedex

Contraintes de deducibilité avec négation

Résumé : Voir "Abstract"

Mots-clés : Services Web, orchestration, politique de sécurité, séparation des tâches, contraintes de deducibilité, protocoles de sécurité, méthodes formelles, util

1 Introduction

1.1 Context

Trust and security management in distributed frameworks is known to be a non-trivial critical issue. It is particularly challenging in Service Oriented Architecture where services can be discovered and composed in a dynamic way. Implemented solutions should meet the seemingly antinomic goals of openness and flexibility on one hand and compliance with data privacy and other regulations on the other hand. We have demonstrated in previous works [6, 22, 2] that functional agility can be achieved for services with a message-level security policy by providing an automated service synthesis algorithm. It resolves a system of deducibility constraints by synthesizing a *mediator* that may adapt, compose and analyze messages exchanged between client services and having the functionalities specified by a goal service. It is complete as long as the security policies only apply to the participants in the orchestration and not on the synthesized service nor on who is able to participate. However security policies often include such *non-deducibility* constraints on the mediator. For instance an organisation may not be trusted to efficiently protect the customer's data against attackers even though it is well-meaning. In this case a client would require that the mediator synthesized to interact with this organization must not have direct access to her private data, which is an effective protection even in case of total compromise. Also it is not possible to specify that the mediator enforces *e.g.* dynamic separation of duty, *i.e.*, restrictions on the possible participants based on the messages exchanged.

Since checking whether a solution computed by our previous algorithm satisfies the non-deducibility constraints is not complete, we propose in this paper to solve during the automated synthesis of the mediator both deducibility and non-deducibility constraints. The former are employed to specify a mediator that satisfies the functional requirements and the security policy on the messages exchanged by the participants whereas the latter are employed to enforce a security policy on the mediator and the participants to the orchestration.

Original contribution. We have previously proposed decision procedures [6, 22, 2] for generating a mediator from a high-level specification with deducibility constraints of a goal service. In this paper we extend the formalism to include non-deducibility constraints in the specification of the mediator and provide a decision procedure synthesizing a mediator for the resulting constraint systems.

Related works. In order to understand and anticipate potential flaws in complex composition scenarios, several approaches have been proposed for the formal specification and analysis of secure services [10, 8]. Among the works dedicated to trust in multi-agent systems, the models closest to ours are [12, 15] in which one can express that an agent trusts another agent in doing or forbearing of doing an action that leads to some goal. To our knowledge no work has previously considered the automatic orchestration of security services with policies altogether as ours. However there are some interesting related attempts to analyze security protocols and trust management [17, 11]. In [17] the author uniformly models security protocols and access control based on trust management. The work introduces an elegant approach to model automated trust negotiation. We also consider an integrated framework for protocols and policies but in our case *i*) policies can be explicitly negative such as non-disclosure policies and separation-of-duty *ii*) we propose a decision procedure for the related trust negotiation problem *iii*) we do not consider indistinguishability properties. In [11] security protocols are combined with authorization logics that can be expressed with acyclic Horn clauses. The authors encode the derivation of authorization predicates (for a service) as subprotocols and can reuse in that way the constraint solving algorithm from [19] to obtain a decision procedure. In our case we consider more general intruder theories (subterm convergent ones) but focus on negation. We conjecture that our approach applies to their authorization policies too.

Our decision procedure for general (negative and positive) constraints extend [7] where negative constraints are limited to have ground terms in right-hand sides, and the deduction system is Dolev-Yao system [9], a special instance of the subterm deduction systems we consider here. In [14] the authors study a class of contract signing protocols where some very specific Dolev-Yao negative constraints are implicitly handled.

Finally one should note that the non deducibility constraints we consider tell that some data cannot be disclosed *globally* but they cannot express finer-grained privacy or information leakage notions relying on probability such as for instance differential privacy.

Paper organization. In Subsection 1.2 we introduce a motivating banking application and sketch our approach to obtain a mediator service. To our knowledge this application is out of the scope of alternative automatic methods. In Section 2 we present our formal setting. A deduction system (Subsection 2.2) describes the abilities of the mediator to process the messages. The mediator synthesis problem is reduced to the resolution of constraints that are defined in Subsection 2. In Section 3 we recall the class of *subterm deduction systems* and their properties. These systems have nice properties that allow us to decide in Section 4 the satisfiability of deducibility constraints even with negation. Finally we conclude in Section 5.

1.2 Synthesis of a Loan Origination Process (LOP)

We illustrate how negative constraints are needed to express elaborated policies such as Separation of Duty by a classical loan origination process example. Our goal is to synthesize a mediator that selects two bank clerks satisfying the Separation of Duty policy to manage the client request. Such a problem is solved automatically by the decision procedure proved in the following sections. Let us walk through the specification of the different parts of the orchestration problem.

Formal setting. Data are represented by first-order terms defined on a signature that comprises binary symbols for symmetric and assymetric encryptions (resp. $\{|\_|\}_{\_}$, $\{\_\}_{\_}$), signature ($\{\_\}^{\mathrm{sig}}_{\_}$), and pairing (pair). Given a public key k we write $\mathrm{inv}\,(k)$ its associated private key. For example $\{a\}^{\mathrm{sig}}_{\mathrm{inv}(k)}$ is the signature of a by the owner of public key k. For readability we write $a.b.c$ a term $\mathrm{pair}\,(a, \mathrm{pair}\,(b, c))$. The binary symbol rel expresses that two agents are related and is used for defining a Separation of Duty policy. A unary symbol g is employed to designate participants identity in the "relatives" database.

Client and clerks. The client and the clerks are specified by services with a security policy, specifying the cryptographic protections and the data and security tokens, and a business logic that specify the sequence in which the operations may be invoked. These are compiled into a sequence of protected messages each service is willing to follow during the orchestration (Fig. 1 and 2).

Client C wants to ask for a loan from a service P, but for this he needs to get an approval from two banking clerks. He declares his intention by sending to mediator M a signed by him message containing service name P and the identity of the client $g(C)$. The mediator should send back the names of two clerks A and B who will evaluate his request. The client then sends to each clerk a request containing amount $Amnt$, his name C and a fresh key N_k which should be used to encrypt decisions. Each request is encrypted with a public key of the corresponding clerk ($pk(A)$ or $pk(B)$). Then the mediator must furnish the decisions (R_a and R_b) of two clerks each encrypted with the proposed key N_k and also their signatures. Finally, the client uses these tokens to ask his loan from P, where $pk(P)$ is a public key of P.

Clerk A receives a request to participate in a LOP which is conducted by mediator M. If he accepts, he returns his identity and public key. Then Clerk receives the client's request for a loan to evaluate: amount Amnt, client's name C and a temporary key K for encrypting his decision. The last is sent back together with a signature certifying the authenticity of this decision on the given request.

The client's non-disclosure policy is given in Fig. 2 and is self-explanatory. Let us explain the services' non-disclosure policy. The Clerk's decision (its last message) should be unforgeable, thus, it should not be known by the Mediator before it was sent by the Clerk (first non-disclosure constraint of Fig. 1). The role clerk played by A can be used by the mediator only if the constraint $\natural g(A)$ is satisfied, showing that A is not a relative with any other actor of the protocol, as client and the other clerk (second non-disclosure constraint of Fig. 1).

Goal service. In contrast with the other services and clients, the goal service is only described in terms of possible operations and available initial data.

Initial data. Beside his private/public keys and the public keys of potential partners (*e.g.* $\mathrm{pk}\,(P)$) the goal service has access to a relational database $\mathrm{rel}(g(a), g(c)), \mathrm{rel}(g(b), g(c)), \ldots$ for storing known existing relations between agents to be checked against conflict of interests.

Clerk's (A) communications:[1]

$* \Rightarrow$ A : request.M
A $\Rightarrow$ M : $g(A).\,\mathrm{pk}\,(A)$
M $\Rightarrow$ A : $\{Amnt.C.K\}_{\mathrm{pk}(A)}$
A $\Rightarrow$ M : $m_1(A, Resp_A, K, C, Amnt)$

Non-disclosure constraints:

1. M cannot deduce the fourth message before it is sent by A.
2. M cannot deduce $g(A)$ before the second message is sent by A.

Figure 1: Clerk's communications and non-disclosure constraints

Client's (C) communications:[1]

C $\Rightarrow$ M : $\{g(C).loan.P\}^{\mathrm{sig}}_{\mathrm{inv}(pk(C))}$
M $\Rightarrow$ C : $A.B$
C $\Rightarrow$ M : $m_2(A, Amnt).m_2(B, Amnt)$
M $\Rightarrow$ C : $m_3(A, R_a).m_3(B, R_b)$
C $\Rightarrow$ P : $m_4(\mathrm{pk}\,(P)\,, A, B, R_a, R_b)$

Non-disclosure constraints:

1. M cannot deduce the amount $Amnt$.
2. M cannot deduce A's decision R_a.
3. M cannot deduce B's decision R_b.

Figure 2: Client's Communications and non-disclosure constraints

Composition rules			Decomposition rules					
x, y	$\rightarrow$	$\mathrm{pair}\,(x, y)$	$\mathrm{pair}\,(x, y)$	$\rightarrow$	x			
			$\mathrm{pair}\,(x, y)$	$\rightarrow$	y	$x, \mathrm{rel}(x, y)$	$\rightarrow$	y
x, y	$\rightarrow$	$\{\lvert x\rvert\}_y$	$y, \{\lvert x\rvert\}_y$	$\rightarrow$	x	$y, \mathrm{rel}(x, y)$	$\rightarrow$	x
x, y	$\rightarrow$	$\{x\}_y$	$\mathrm{inv}\,(y)\,, \{x\}_y$	$\rightarrow$	x			
$x, \mathrm{inv}\,(y)$	$\rightarrow$	$\{x\}^{\mathrm{sig}}_{\mathrm{inv}(y)}$	$y, \{x\}^{\mathrm{sig}}_{\mathrm{inv}(y)}$	$\rightarrow$	x			

Figure 3: Deduction system for the LOP example.

Deduction rules. The access to the database as well as the possible operations on messages are modeled by a set of deduction rules (formally defined later). We anticipate on the rest of this paper, and present the rules specific to this case study grouped into composition and decomposition rules in Fig. 3.

Mediator synthesis problem. In order to communicate with the services (here the client, the clerks and the service P), a mediator has to satisfy a sequence of constraints expressing that *(i)* each message m expected by a service (denoted $?m$) can be deduced from all the previously sent messages m' (denoted $!m'$) and the initial knowledge and *(ii)* each message w that should not be known or disclosed (denoted $\natural w$ and called negative constraint) is not deducible.

The orchestration problem consists in finding a satisfying interleaving of the constraints imposed by each service. For instance, clerk's and client's constraints extracted from Fig. 1 and Fig. 2 are:

$$\begin{cases} Client(C) \triangleq & !_M \{g(C).loan.P\}^{\mathrm{sig}}_{\mathrm{inv}(K_C)}\ ?_M A.B\ !_M m_2(A, Amnt).m_2(B, Amnt) \\ & ?_M m_3(A, R_a).m_3(B, R_b)\ \natural_M Amnt\ \natural_M R_A\ \natural_M R_B \\ & !_P m_4(\mathrm{pk}(P), A, B, R_a, R_b) \\ Clerk(A) \triangleq & ?request.M\ \natural_M g(A)\ !_M g(A).\mathrm{pk}(A)\ ?_M \{Amnt.C.K\}_{\mathrm{pk}(A)} \\ & \natural_M m_1(A, Resp_A, K, C, Amnt)\ !_M m_1(A, Resp_A, K, C, Amnt) \end{cases}$$

If it exists our procedure outputs a solution which can be translated automatically into a mediator. Note, for example, that without the negative constraint $\natural g(A)$ a synthesized mediator might accept any clerk identity and that could violate the Separation of Duty policy.

2 Derivations and constraint systems

In our setting messages are terms generated or obtained according to some elementary rules called *deduction rules.* A *derivation* is a sequence of deduction rules applied by a mediator to build new messages. The goal of the synthesis is specified by a *constraint system*, *i.e.* a sequence of terms labelled by symbols !,? or $\natural$, respectively *sent*, *received*, or *unknown* at some step of the process.

2.1 Terms and substitutions

Let $\mathcal{X}$ be a set of *variables*, $\mathcal{F}$ be a set of *function symbols* and $\mathcal{C}$ a set of *constants*. The set of *terms* $\mathcal{T}$ is the minimal set containing $\mathcal{X}$, $\mathcal{C}$ and if $t_1, \ldots, t_k \in \mathcal{T}$ then $f(t_1, \ldots, t_k) \in \mathcal{T}$ for any $f \in \mathcal{F}$ with arity k. The set of *subterms* of a term t is denoted $\mathrm{Sub}(t)$ and is the minimal set containing t such that $f(t_1, \ldots, t_n) \in \mathrm{Sub}(t)$ implies $t_1, \ldots, t_n \in \mathrm{Sub}(t)$ for $f \in \mathcal{F}$. We denote $\mathrm{Vars}(t)$ the set $\mathcal{X} \cap \mathrm{Sub}(t)$. A term t is *ground* is $\mathrm{Vars}(t) = \emptyset$. We denote $\mathcal{T}_g$ the set of ground terms.

A *substitution* σ is an idempotent mapping from $\mathcal{X}$ to $\mathcal{T}$. It is ground if it is a mapping from $\mathcal{X}$ to $\mathcal{T}_g$. The application of a substitution σ on a term t is denoted $t\sigma$ and is equal to the term t where all variables x have been replaced by the term $x\sigma$. We say that a substitution σ is *injective* on a set of terms T, iff for all $p, q \in T$ $p\sigma = q\sigma$ implies $p = q$. The *domain* of σ (denoted by $\mathrm{dom}(\sigma)$) is set: $\{x \in \mathcal{X} : x\sigma \neq x\}$. The *image* of σ is $\mathrm{img}(\sigma) = \{x\sigma : x \in \mathrm{dom}(\sigma)\}$. Given two substitutions σ, δ, the substitution $\sigma\delta$ has for domain $\mathrm{dom}(\sigma) \cup \mathrm{dom}(\delta)$ and is defined by $x\sigma\delta = (x\sigma)\delta$. If $\mathrm{dom}(\sigma) \cap \mathrm{dom}(\delta) = \emptyset$ we write $\sigma \cup \delta$ instead of $\sigma\delta$.

A *unification system* U is a finite set of equations $\{p_i =_? q_i\}_{1 \le i \le n}$ where $p_i, q_i \in \mathcal{T}$. A substitution σ is an *unifier* of U or equivalently satisfies U iff for

[1] We have employed the following abbreviations for messages:

$$\begin{cases} m_1(A, Resp, K, Ct, S) & = & \{h(A.S.Ct.Resp)\}^{\mathrm{sig}}_{\mathrm{inv}(pk(A))}.\{|Resp|\}_K \\ m_2(A, S) & = & \{S.C.N_k\}_{\mathrm{pk}(A)} \\ m_3(A, R) & = & m_1(A, R, N_k, C, Amnt) \\ m_4(K_0, A, B, R_1, R_2) & = & \{Amnt.C.A.R_1.B.R_2\}_{K_0}.m_3(A, R_1).m_3(B, R_2) \end{cases}$$

all $i = 1, \ldots, n$, $p_i\sigma = q_i\sigma$. Any satisfiable unification system U admits a *most general unifier* $\mathrm{mgu}(U)$, unique modulo variable renaming, and such that for any unifier σ of U there exists a substitution τ such that $\sigma = \mathrm{mgu}(U)\,\tau$. Wlog we assume in the rest of this paper that $\mathrm{Vars}(\mathrm{img}(\mathrm{mgu}(U))) \subseteq \mathrm{Vars}(U)$, *i.e.*, the most general unifier does not introduce new variables.

A *sequence* s is indexed by $[1, \ldots, n]$ with $n \in \mathbb{N}$. We write $|s|$ the length of s, $\emptyset$ the empty sequence, $s[i]$ the ith element of s, $s[m:n]$ the sequence $s[m], \ldots, s[n]$ and s, s' the concatenation of two sequences s and s'. We write $e \in s$ and $E \subseteq s$ for, respectively, $\exists i : s[i] = e$ and $\forall e \in E, e \in s$.

2.2 Deduction systems

The new values created by the mediator are constants in a subset $\mathcal{C}_{\mathrm{med}}$ of $\mathcal{C}$. We assume that both $\mathcal{C}_{\mathrm{med}}$ and $\mathcal{C} \setminus \mathcal{C}_{\mathrm{med}}$ are infinite. Given $l_1, \ldots, l_n, r \in \mathcal{T}$, the notation $l_1, \ldots, l_n \to r$ denotes a *deduction rule* if $\mathrm{Var}(r) \subseteq \bigcup_{i=1}^{n} \mathrm{Var}(l_i)$. A *deduction* is a ground instance of a deduction rule. A *deduction system* is a set of deduction rules that contains a finite set of deduction rules in addition to all *nonce creation rules* $\to n$ (one for every $n \in \mathcal{C}_{\mathrm{med}}$) and all reception rules $?t$ (one for every $t \in \mathcal{T}$). All rules but the reception rules are called *standard* rules. The deduction system describes the abilities of the mediator to process the messages. In the rest of this section we fix an arbitrary deduction system $\mathcal{D}$. We denote by $l \twoheadrightarrow\!\!\!\!\!\!\;\shortmid\; r$ any rule and $l \to r$ any standard rule.

2.3 Derivations and localizations

A *derivation* is a sequence of deductions, including receptions of messages from available services, performed by the mediator. Given a sequence of deductions $E = (l_i \twoheadrightarrow\!\!\!\!\!\!\;\shortmid\; r_i)_{i=1,\ldots,m}$ we denote $\mathrm{R}_E\,(i)$ the set $\{r_j \,:\, j \leq i\}$.

Definition 2.1 (Derivation)**.** *A sequence of deductions* $D = (l_i \twoheadrightarrow\!\!\!\!\!\!\;\shortmid\; r_i)_{i=1,\ldots,m}$ *is a* derivation *if for any* $i \in \{1, \ldots, m\}$*,* $l_i \subseteq \mathrm{R}_D\,(i-1)$*.*

Given a derivation D we define $\mathrm{Next}_D(i) = \min(\{|D|+1\} \cup \{j : j > i \text{ and } D[j] = ?t_j\})$. The explicit knowledge of the mediator is the set of terms it has already deduced, and its implicit knowledge is the set of terms it can deduce. If the former is K we denote the latter $\mathrm{Der}(K)$. A derivation D is a *proof* of $s \in \mathrm{Der}(K)$ if $?r \in D$ implies $r \in K$, and $D[|D|] = l \twoheadrightarrow\!\!\!\!\!\!\;\shortmid\; t$. Thus, we have:

$$\mathrm{Der}(K) = \{t \,:\, \exists D \text{ derivation s.t. } ?r \in D \text{ implies } r \in K, \text{ and } D[|D|] = l \to t\}$$

2.4 Constraint systems

Definition 2.2 (Constraint system)**.** *A constraint system $\mathcal{S}$ is a sequence of constraints where each constraint has one of three forms (where t is a term):*

1. *$?t$, denoting a message reception by an available service or a client,*
2. *$!t$, denoting a message emission by an available service or a client,*

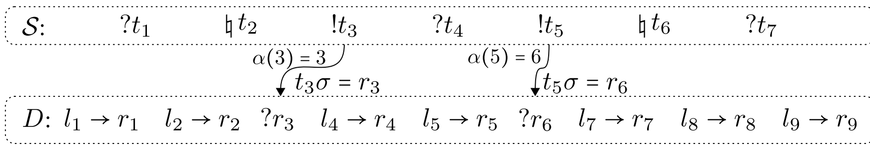

Figure 4: A constraint system and a compliant derivation

3. $\natural t$*, a negative constraint, denoting that the mediator must not be able to deduce* t *at this point;*

and that satisfies the following properties for any $1 \leq i \leq |\mathcal{S}|$*:*

Origination: *if* $\mathcal{S}[i] = !t_i$ *then* $\mathrm{Vars}\,(t_i) \subseteq \bigcup_{j<i} \mathrm{Vars}\,(\{t_j \,:\, \mathcal{S}[j] = ?t_j\})$*;*

Determination: *if* $\mathcal{S}[i] = \natural t_i$ *then* $\mathrm{Vars}\,(t_i) \subseteq \bigcup_{j} \mathrm{Vars}\,(\{t_j \,:\, \mathcal{S}[j] = ?t_j\})$*.*

Origination means that every unknown in a service's state originates from previous input by the mediator. *Determination* means that negative constraints are on messages determined by a service's state at the end of its execution.

In the rest of this paper $\mathcal{S}$ (and decorations thereof) denotes a constraint system. An index i is a *send* (*resp.* a *receive*) index if $\mathcal{S}[i] = !t$ (*resp.* $\mathcal{S}[i] = ?t$) for some term t. If $i_1, \ldots, i_k$ is the sequence of all send (*resp.* receive) indices in $\mathcal{S}$ we denote $\mathrm{Out}(\mathcal{S})$ (*resp.* $\mathrm{In}(\mathcal{S})$) the sequence $\mathcal{S}[i_1], \ldots, \mathcal{S}[i_k]$. We note that the origination and determination properties imply $\mathrm{Var}(\mathcal{S}) = \mathrm{Var}(\mathrm{In}(\mathcal{S}))$. Given $1 \leq i \leq |\mathcal{S}|$ we denote $\mathrm{prev}_{\mathcal{S}}(i)$ to be $\max(\{0\} \cup \{j \,:\, j \leq i \text{ and } \mathcal{S}[j] = !t_j\})$.

Definition 2.3 (Solution of a constraint system)**.** *A ground substitution* σ *is a solution of* $\mathcal{S}$*, and we denote* $\sigma \models \mathcal{S}$*, if* $\mathrm{dom}\,(\sigma) = \mathrm{Var}(\mathcal{S})$ *and*

1. *if* $\mathcal{S}[i] = ?t$ *then* $t\sigma \in \mathrm{Der}(\{t_j\sigma \,:\, j \leq \mathrm{prev}_{\mathcal{S}}(i) \text{ and } \mathcal{S}[j] = !t_j\})$
2. *if* $\mathcal{S}[i] = \natural t$ *then* $t\sigma \notin \mathrm{Der}(\{t_j\sigma \,:\, j \leq \mathrm{prev}_{\mathcal{S}}(i) \text{ and } \mathcal{S}[j] = !t_j\})$

Definition 2.4 (Compliant derivations)**.** *Let* σ *be a ground substitution with* $\mathrm{dom}\,(\sigma) = \mathrm{Var}(\mathcal{S})$*. A derivation* D *is* $(\mathcal{S}, \sigma)$-compliant *if there exists a strictly increasing bijective mapping* α *from the send indices of* $\mathcal{S}$ *to the set* $\{j \,:\, D[j] = ?r\}$ *such that* $\mathcal{S}[i] = !t$ *implies* $D[\alpha(i)] = ?t\sigma$*.*

An example of $(\mathcal{S}, \sigma)$-compliant derivation is shown in Figure 4. Since a sequence of receptions is a derivation, we note that for every ground substitution σ with $\mathrm{dom}\,(\sigma) = \mathrm{Var}(\mathrm{In}(\mathcal{S}))$ there exists at least one compliant derivation D.

Definition 2.5 (Proof of a solution)**.** *Let* σ *be a ground substitution. A derivation* D *is a* proof *of* $\sigma \models \mathcal{S}$*, and we denote* $D, \sigma, \alpha \vdash \mathcal{S}$*, if:*

1. D *is* $(\mathcal{S}, \sigma)$*-compliant with the mapping* α *and*
2. *if* $\mathcal{S}[i] = ?t$ *there is* $j < \mathrm{Next}_D(\alpha(\mathrm{prev}_{\mathcal{S}}(i)))$ *such that* $D[i] = l \twoheadrightarrow\!\!\!\!\!\!/\;\; t\sigma$ *and*
3. *if* $\mathcal{S}[i] = \natural t$ *then* $t\sigma \notin \mathrm{Der}(\{t_j\sigma \,:\, j \leq \mathrm{prev}_{\mathcal{S}}(i) \text{ and } \mathcal{S}[j] = !t_j\})$.

In Figure 4, if σ is a solution of $\mathcal{S}$ and, for example, $t_1\sigma = r_2$, $t_2\sigma \notin \mathrm{Der}(\emptyset)$, $t_4\sigma = r_4$, $t_6\sigma \notin \mathrm{Der}(\{r_3, r_6\})$ and $t_7\sigma = r_8$ then D is a proof of $\sigma \models \mathcal{S}$.

Let us prove that if $\sigma \models \mathcal{S}$ then there is a proof $D, \sigma, \alpha \vdash \mathcal{S}$.

Definition 2.6 (Maximal derivation)**.** *Let T be a finite set of terms and σ be a ground substitution with* $\mathrm{dom}\,(\sigma) = \mathrm{Var}(T)$*. A derivation D is (T, σ)-maximal iff for every $t \in \mathrm{Sub}(T)$, $t\sigma \in \mathrm{Der}(\mathrm{R}_D\,(i))$ implies $t\sigma \in \mathrm{R}_D\,(\mathrm{Next}_D(i) - 1)$.*

First we prove that maximal derivations are natural proof candidates of $\sigma \models \mathcal{S}$.

Lemma 1. *Let σ be a ground substitution with* $\mathrm{dom}\,(\sigma) = \mathrm{Var}(\mathcal{S})$ *and D be a $(\mathcal{S}, \sigma)$-compliant* $(\mathrm{Sub}(\mathcal{S}), \sigma)$*-maximal derivation.* ***Then*** *$\sigma \models \mathcal{S}$ iff for all i*

- *if $\mathcal{S}[i] = ?t$ then there exists $j < \mathrm{Next}_D(\alpha(\mathrm{prev}_{\mathcal{S}}(i))) : D[j] = l \twoheadrightarrow t\sigma$ and*
- *if $\mathcal{S}[i] = \natural t$ then for all $j < \mathrm{Next}_D(\alpha(\mathrm{prev}_{\mathcal{S}}(i))) : D[j] \neq l \twoheadrightarrow t\sigma$.*

In the next lemma we show that any (T, σ)-maximal derivation D may be extended into a (T', σ')-maximal derivation for an arbitrary extension T', σ' of T, σ by adding into D only standard deductions.

Lemma 2. *Let σ be a ground substitution with* $\mathrm{dom}\,(\sigma) = \mathrm{Var}(\mathcal{S})$*. Let T_1, T_2 be two sets of terms such that $T_1 \subseteq T_2$, and σ_1, σ_2 be two substitutions such that* $\mathrm{dom}\,(\sigma_1) = \mathrm{Var}(T_1)$ *and* $\mathrm{dom}\,(\sigma_2) = \mathrm{Var}(T_2) \setminus \mathrm{Var}(T_1)$*. If D is a (T_1, σ_1)-maximal $(\mathcal{S}, \sigma)$-compliant derivation in which no term is deduced twice by a standard rule,* ***then*** *there exists a $(T_2, \sigma_1 \cup \sigma_2)$-maximal $(\mathcal{S}, \sigma)$-compliant derivation D' in which no term is deduced twice by a standard rule such that every deduction whose right-hand side is in* $\mathrm{Sub}(T_1)\sigma_1$ *occurs in D' iff it occurs in D.*

Proof. Let $i_1, \ldots, i_k$ be the indices of the non-standard rules in D, let $D[i_j] = ?t_{i_j}$, and let for $0 \leq j \leq k$ $D_j = D[i_j{+}1 : i_{j+1}{-}1]$ with $i_0 = 0$ and $i_{k+1} = |D|{+}1$. That is, $D = D_0, !t_{i_1}, D_1, !t_{i_2}, D_2, \ldots !t_{i_k}, D_k$. Noting that $\mathrm{dom}\,(\sigma_1) \cap \mathrm{dom}\,(\sigma_2) = \emptyset$ let $\sigma' = \sigma_1 \cup \sigma_2$.

For each $t \in \mathrm{Sub}(T_2)$ such that $t\sigma' \in \mathrm{Der}(t_{i_1}, \ldots, t_{i_k})$ let i_t be minimal such that $t\sigma' \in \mathrm{Der}(t_{i_1}, \ldots, t_{i_t})$, and let E_t^0 be a proof of this fact, and E_t be a sequence of standard deductions obtained by removing every non-standard deduction from E_t^0.

For $0 \leq j \leq k$ let D'_j be the sequence of standard deduction steps $D_j, E_{s_1}, \ldots, E_{s_p}$ for all $s_m \in \mathrm{Sub}(T_2)\sigma' \setminus \mathrm{Sub}(T_1)\sigma'$ such that $i_{s_m} = j$ in which every rule of $E_{s_1}, \ldots, E_{s_p}$ that deduces a term previously deduced in the sequence or for some $m \leq j$ deduced in D'_m or in $D[i_m]$ is removed.

Let $D' = D'_0, ?t_{i_1}, D'_1, \ldots, ?t_{i_k}, D'_k$. We have deleted in each E_t^0 only deductions whose right-hand side occurs before in D', and thus D' is a derivation. Since the D'_i contains only standard deductions, we can see that D' is $(\mathcal{S}, \sigma)$-compliant.

Since D is (T_1, σ_1)-maximal and no term is deduced twice in D we note that, for $t \in T_1$, no standard deduction of $t\sigma_1$ from a sequence D_j is deleted. Furthermore we note that standard deductions of terms $T_2\sigma_2$ that are also in

$T_1\sigma_1$ are deleted by construction and by the maximality of D. Thus a deduction whose right-hand side is in $\mathrm{Sub}(T_1)\sigma_1$ is in D' iff it occurs in D.

By construction D' is (T_2, σ')-maximal and no term is deduced twice by standard deductions. □

Taking $T_1 = \emptyset$, $T_2 = \mathrm{Sub}(\mathcal{S})$, and $\sigma_2 = \sigma$, Lemma 2 implies that for every substitution σ of domain $\mathrm{Var}(\mathcal{S})$ there exists a $(\mathcal{S}, \sigma)$-compliant $(\mathrm{Sub}(\mathcal{S}), \sigma)$-maximal derivation D. By Lemma 1 if $\sigma \models \mathcal{S}$ then D is a proof of $\sigma \models \mathcal{S}$. Since the converse is trivial, it suffices to search proofs maximal wrt $T \supseteq \mathrm{Sub}(\mathcal{S})$.

3 Subterm deduction system

3.1 Definition and main property

We say that a deduction system is a *subterm deduction system* whenever each deduction rule which is not a nonce creation or a message reception is either:

1. $x_1, \ldots, x_n \to f(x_1, \ldots, x_n)$ for a function symbol f;
2. $l_1, \ldots, l_n \to r$ for some terms $l_1, \ldots, l_n, r$ such that $r \in \bigcup_{i=1}^{n} \mathrm{Sub}(l_i)$.

A *composition* rule is either a message reception, a nonce creation, or a rule of the first type. A deduction rule is otherwise a *decomposition* rule. Reachability problems for deduction systems with a convergent equational theory are reducible to the satisfiability of a constraint system in the empty theory for a deduction system in our setting [16, 13]. If furthermore the equational theory is *subterm* [5] the reduction is to a subterm deduction system as just defined above.

Now we show that if $D, \sigma, \alpha \vdash \mathcal{S}$, a term $s \in \mathrm{Sub}(D)$ is either the instance of a non-variable subterm of $\mathrm{Out}(\mathcal{S})$ or deduced by a standard composition.

Lemma 3. *Let σ be a ground substitution such that $\sigma \models \mathcal{S}$. If D is a proof of $\sigma \models \mathcal{S}$ such that no term is deduced twice in D by standard rules and s is a term such that $s \in \mathrm{Sub}(D)$ and $s \notin (\mathrm{Sub}(\mathrm{Out}(\mathcal{S})) \setminus \mathcal{X})\sigma$* ***then*** *there exists an index i in D such that $D[i] = l \to s$ is a composition rule and $s \notin \mathrm{Sub}(\mathrm{R}_D\,(i-1))$.*

Proof. First we note that by definition of subterm deduction systems for any decomposition rule $l \to r$ we have *a)* $r \in \mathrm{Sub}(l)$, and *b)* for any composition rule $l \to r$ we have $l \subset \mathrm{Sub}(r)$ and $\mathrm{Sub}(r) \setminus \mathrm{Sub}(l) = \{r\}$.

Let D be a proof of $\sigma \models \mathcal{S}$, and let i be minimal such that $D[i] = l_r \twoheadrightarrow r$ with $s \in \mathrm{Sub}(r)$. Since $l_r \subseteq \mathrm{R}_D\,(i-1)$, the minimality of i implies $s \in \mathrm{Sub}(r) \setminus \mathrm{Sub}(l_r)$.

Thus by *a)* $D[i]$ cannot be a decomposition.

If $D[i] = ?r$ then by the $(\mathcal{S}, \sigma)$-compliance of D we have $\mathcal{S}[\alpha^{-1}(i)] = !t$ with $t\sigma = r$. We have $s \in \mathrm{Sub}(r) = \mathrm{Sub}(t\sigma) = \mathrm{Sub}(t)\sigma \cup \mathrm{Sub}(\mathrm{Vars}\,(t)\,\sigma)$.

If $s \in (\mathrm{Sub}(\mathrm{Out}(S)) \setminus \mathcal{X})\sigma$ we are done, otherwise there exists $y \in \mathrm{Vars}\,(t)$ with $s \in \mathrm{Sub}(y\sigma)$. By the origination property, there exists $k < \alpha^{-1}(i)$ such that $\mathcal{S}[k] = ?t'$ with $y \in \mathrm{Vars}\,(t')$. Since $D, \sigma, \alpha \vdash \mathcal{S}$ and $k < \alpha^{-1}(i)$ there exists $j < i$ such that $D[j] = l_j \to t'\sigma$. The minimality of i is contradicted by $s \in \mathrm{Sub}(t'\sigma)$.

Therefore, $D[i] = l_r \to r$ is a standard composition rule. As a consequence, $\mathrm{Sub}(r) \setminus \mathrm{Sub}(l_r) = \{r\}$. Since $s \in \mathrm{Sub}(r) \setminus \mathrm{Sub}(l_r)$, we finally obtain $s = r$. □

3.2 Locality

Subterm deduction systems are not necessarily local in the sense of [18]. However we prove in this subsection that given σ, there exists a finite extension T of $\mathrm{Sub}(\mathcal{S})$ and an extension σ' of σ of domain $\mathrm{Var}(T)$ and a (T, σ')-maximal derivation D in which every deduction relevant to the proof of $\sigma \models \mathcal{S}$ is liftable into a deduction between terms in T. Let us first precise the above statements.

Definition 3.1 (Localization set)**.** *A set of terms T* localizes *a derivation $D = (l_i \twoheadrightarrow r_i)_{1 \leq i \leq m}$ for a substitution σ of domain* $\mathrm{Var}(T)$ *if for every $1 \leq i \leq m$ if $D[i]$ is a standard rule and there exists $t \in \mathrm{Sub}(T) \setminus \mathcal{X}$ such that $t\sigma = r_i$, there exists $t_1, \ldots, t_n \in \mathrm{Sub}(T)$ such that $\{t_1\sigma, \ldots, t_n\sigma\} \subseteq \mathrm{R}_D\,(i-1)$ and $t_1, \ldots, t_n \to t$ is the instance of a standard deduction rule.*

First, we prove that for subterm deduction systems, every proof D of $\sigma \models \mathcal{S}$ is localized by a set T of DAG size linear in the DAG size of $\mathcal{S}$.

Lemma 4. *If σ is a ground substitution such that $\sigma \models \mathcal{S}$ there exists $T \supseteq$* $\mathrm{Sub}(\mathcal{S})$ *of size linear in* $|\mathrm{Sub}(\mathcal{S})|$*, a substitution τ of domain* $\mathrm{Var}(T) \setminus \mathrm{Var}(\mathcal{S})$ *and a $(T, \sigma \cup \tau)$-maximal and $(\mathcal{S}, \sigma)$-compliant derivation localized by T for $\sigma \cup \tau$.*

Proof. By Lemma 2 applied with $T_1 = \emptyset$, $T_2 = \mathrm{Sub}(\mathcal{S})$, $\sigma_1 = \emptyset, \sigma_2 = \sigma$, and D_0 the $(\mathcal{S}, \sigma)$-compliant derivation that has no standard deductions, there exists a $(\mathrm{Sub}(\mathcal{S}), \sigma)$-maximal $(\mathcal{S}, \sigma)$-compliant derivation D in which no term is deduced twice by a standard deduction. From now on we let $T_0 = \mathrm{Sub}(\mathcal{S})$.

Let $\{l_i \to r_i\}_{1 \leq i \leq n}$ be the set of decompositions in D, and $\{(L_i \to R_i, \tau_i)\}_{1 \leq i \leq n}$ be a set of decomposition rules and ground substitutions such that for all $1 \leq i \leq n$ we have $L_i\tau_i \to R_i\tau_i = l_i \to r_i$. Since no term in D is deduced twice by a standard deduction, by Lemma 3 we have $n \leq |\mathrm{Sub}(\mathrm{Out}(S))|$.

Modulo variable renaming we may assume that $i \neq j$ implies $\mathrm{dom}\,(\tau_i) \cap \mathrm{dom}\,(\tau_j) = \emptyset$, and thus that $\tau = \bigcup_{i=1}^{n} \tau_i$ is defined on $T_1 = \bigcup_{i=1}^{n}(\mathrm{Sub}(L_i) \cup \mathrm{Sub}(R_i))$. Note that the size of T_1 is bounded by $M \times |\mathrm{Sub}(\mathrm{Out}(\mathcal{S}))|$, where M is the maximal size of a decomposition rule belonging to the deduction system.

Let $T = T_0 \cup T_1$ and, noting that these substitutions are defined on non-intersecting domains, let $\sigma' = \sigma \cup \tau$. By construction $|T| \leq (M+1) \times |\mathrm{Sub}(\mathcal{S})|$.

By Lemma 2 there exists a $(\mathcal{S}, \sigma)$-compliant derivation D' which is (T, σ')-maximal and such that every deduction of a term in $T_0\sigma$ that occurs in D also occurs in D' and no term is deduced twice in D' by a standard deduction.

Let $l \to r$ be a deduction in D' which does not appear in D. Since D is (T_0, σ)-maximal we have $r \notin \mathrm{Sub}(T_0)\sigma$, and thus $r \notin \mathrm{Sub}(\mathrm{Out}(\mathcal{S}))\sigma$. Since no term is deduced twice in D' by Lemma 3 this deduction must be a composition.

Let us prove D' is (T, σ')-localized. By definition of composition rules, every composition that deduces a term $t\sigma'$ with $t \in \mathrm{Sub}(T) \setminus \mathrm{Var}(T)$ has a left-hand side $t_1\sigma', \ldots, t_k\sigma'$ with $t_1, \ldots, t_k \in \mathrm{Sub}(T)$ and $t_1, \ldots, t_k \to t$ is an instance of a composition rule. By the preceding paragraph every decomposition in D' occurs in D and thus by construction has its left-hand side in $T_1\sigma'$ which was previously built in D and is an instance of some $L_i \to R_i$ such that $\mathrm{Sub}(L_i \cup \{R_i\}) \subseteq T_1 \subseteq T$.

Thus every deduction whose right-hand side is in $(\mathrm{Sub}(T) \setminus \mathrm{Var}(T))\sigma'$ has its left-hand side in $\mathrm{Sub}(T)\sigma'$, and thus D' is localized by T for σ'. $\square$

We prove now that to solve constraint systems one can first guess equalities between terms in T and then solve constraint systems without variables. The guess of equalities is correct wrt a solution σ if terms in T that have the same instance by σ are syntactically equal. We characterize these guesses as follows.

Definition 3.2 (One-to-one localizations). *A set of terms T* one-to-one localizes *a derivation D for a ground substitution σ if σ is injective on* $\mathrm{Sub}(T)$ *and T localizes D for σ.*

In Lemma 7 we prove that once equalities between variables are correctly guessed there exists a one-to-one localization of a maximal proof D.

Lemma 5. *Let T be a set of terms such that $T = \mathrm{Sub}(T)$, σ be a ground substitution defined on* $\mathrm{Vars}(T)$, *$U = \{p =_? q : p, q \in T \wedge p\sigma = q\sigma\}$ be a unification system and θ be its most general idempotent unifier with* $\mathrm{Vars}(\mathrm{img}(\theta)) \subseteq \mathrm{Vars}(U)$. *Then for any term t, $t\theta\sigma = t\sigma$.*

Proof. Let us show $\forall x \in \mathrm{Vars}(T), x\sigma = x\theta\sigma$. Note that this trivially holds if $x\theta = x$. Thus we consider case $x\theta \neq x$.

Since U contains all equations $p =_? p$ for $p \in \mathrm{Sub}(T) = T$, we have $\mathrm{Sub}(T) = \mathrm{Sub}(U)$. From the idempotency of θ ($\forall y \in \mathrm{Vars}(U), y\theta\theta = y\theta$), we get $\forall y \in \mathrm{Vars}(\mathrm{img}(\theta)), y\theta = y$.

As σ is evidently a unifier of U, there exists a substitution τ such that $\sigma = \theta\tau$ Therefore, $y\sigma = y\theta\tau = y\tau$, i.e. $y\sigma = y\tau$ for all $y \in \mathrm{Vars}(\mathrm{img}(\theta))$. Thus, for any $x \in \mathrm{Vars}(T)$, $x\theta\sigma = x\theta\tau = x\sigma$.

Consequently, for any term t we have $t\sigma = t\theta\sigma$. $\square$

Lemma 6. *Let U be a unification system and $\theta = \mathrm{mgu}(U)$ an idempotent most general unifier with* $\mathrm{Vars}(\mathrm{img}(\theta)) \subseteq \mathrm{Vars}(U)$. *Then $\forall p \in \mathrm{Sub}(\mathrm{img}(\theta))$ $\exists q \in \mathrm{Sub}(U) : p = q\theta$.*

Proof. The case where $p \in \mathcal{X}$ is trivial, since $\mathrm{Vars}(\mathrm{img}(\sigma)) \subseteq \mathrm{Vars}(U)$ and we can take $q = p$. Otherwise, suppose that $p \in \mathrm{Sub}(x\theta | x \in \mathrm{dom}(\theta)) \setminus \mathcal{X}$ is such that $\forall q \in \mathrm{Sub}(U)\ p \neq q\theta$. Let z be a fresh variable. Let $\theta' = \{x \mapsto (x\theta)|_{p \leftarrow z} : x \in \mathrm{dom}(\theta)\}$. Let us denote the height of a term t by $\mathrm{ht}(t)$, and a subterm of t at position l by $t[l]$ and the set of all positions in t by $t[]$.

Let us prove that $\forall u, v \in \mathrm{Sub}(U)\ \ u\theta = v\theta \implies u\theta' = v\theta'$.

- If $u, v \in \mathcal{X}$ then the statement is true by definition.

- If, w.l.o.g., $u \in \mathcal{X}$ but $v \notin \mathcal{X}$. Then for any $l \in v[]$ we have $(u\theta)[l] = (v[l])\theta$. Since $v[l] \in \mathrm{Sub}(U)$ we get $(u\theta)[l] \neq p$ (as we took such p that $\forall q \in \mathrm{Sub}(U)$ $p \neq q\theta$), and therefore, $(v[l])\theta \neq p$. Thus, $(v\theta)|_{p \hookleftarrow z} = v\theta'$. Therefore, $u\theta' = (u\theta)|_{p \hookleftarrow z} = (v\theta)|_{p \hookleftarrow z} = v\theta'$.

- If $u = f(u_1, \ldots, u_k) \wedge v = g(v_1, \ldots, v_m)$ then $f = g$, $m = k$ and for all $i \leq k$ we have $v_i\theta = u_i\theta$. It is enough to prove that $v_i\theta' = u_i\theta'$. Let us prove this case by induction on $\min(\mathrm{ht}\,(u)\,,\mathrm{ht}\,(v))$. For the basis of induction, we have that either $u_i \in \mathcal{X}$ or $v_i \in \mathcal{X}$ (otherwise the basis is not minimal) and we have proved already that $v_i\theta = u_i\theta \implies v_i\theta' = u_i\theta'$. Suppose the statement is true for $\min(\mathrm{ht}\,(u)\,,\mathrm{ht}\,(v)) \leq n$. For $\min(\mathrm{ht}\,(u)\,,\mathrm{ht}\,(v)) = n+1$ we have $\mathrm{ht}\,(u_i\theta) \leq n$, $\mathrm{ht}\,(v_i\theta) \leq n$ and $u_i\theta = v_i\theta$ for all i. Then by induction supposition and two cases considered before we have $u_i\theta' = v_i\theta'$.

Thus, $\forall u, v \in \mathrm{Sub}(U) u\theta = v\theta \implies u\theta' = v\theta'$, i.e. θ' is a unifier of U. Moreover, for all $x \in \mathrm{dom}\,(\theta)\,, x\theta = (x\theta')\gamma$, where $\gamma = \{z \mapsto p\}$.

Since θ is a most general unifier, we have $p \in \mathcal{X}$ which contradicts to $p \in \mathrm{Sub}(x\theta | x \in \mathrm{dom}\,(\theta)) \setminus \mathcal{X}$.

$\square$

Lemma 7. *Let $\mathcal{S}$ be a constraint system, σ be a ground substitution such that $\sigma \models \mathcal{S}$.*

Then there exists a set of terms T, a substitution τ of domain $\mathrm{Var}(T) \setminus \mathrm{Var}(\mathcal{S})$, a substitution θ and a $(\mathcal{S}\theta, \sigma)$-compliant derivation D such that

- *D is $(T, \sigma \cup \tau)$-maximal and one-to-one localized by T for $\sigma \cup \tau$*
- $\sigma \cup \tau = \theta(\sigma \cup \tau)$
- $\mathrm{Sub}(\mathcal{S}\theta) \subseteq T$
- *T and θ of size linear in* $|\mathrm{Sub}(\mathcal{S})|$

Proof. Under the same assumptions, by Lemma 4, there exists $T_0 \supseteq \mathrm{Sub}(\mathcal{S})$ of size linear in $|\mathrm{Sub}(\mathcal{S})|$ and τ of domain $\mathrm{Var}(T_0) \setminus \mathrm{Var}(\mathcal{S})$ such that there exists a $(T_0, \sigma \cup \tau)$-maximal and $(\mathcal{S}, \sigma)$-compliant derivation D which is localized by T_0 for the same substitution $\sigma' = \sigma \cup \tau$.

Let $\mathcal{U} = \{t =_? t' \,:\, t, t' \in \mathrm{Sub}(T_0) \text{ and } t\sigma' = t'\sigma'\}$. The unification system $\mathcal{U}$ has a unifier σ' and thus has a most general solution θ. By Lemma 5, $\sigma' = \theta\sigma'$.

Let $T = \mathrm{Sub}(T_0)\theta$.

Since $\mathrm{Sub}(\mathcal{S}) \subseteq T_0$ we have $\mathrm{Sub}(\mathcal{S}\theta) \subseteq \mathrm{Sub}(T_0\theta)$. Since θ is a most general unifier of $\mathcal{U}$ and $\mathrm{Sub}(\mathcal{U}) = \mathrm{Sub}(T_0)$ we have $\mathrm{Sub}(T_0\theta) = \mathrm{Sub}(T_0)\theta$ by Lemma 6. This implies *(i)* $\mathrm{Sub}(\mathcal{S}\theta) \subseteq T$, *(ii)* θ is of linear size on $|\mathrm{Sub}(T_0)|$ and thus on $|\mathrm{Sub}(\mathcal{S})|$, and *(iii)* T is of linear size on $|\mathrm{Sub}(\mathcal{S})|$. Moreover, as $\sigma' = \theta\sigma'$ we have $\mathrm{Sub}(T)\sigma' = \mathrm{Sub}(T_0)\sigma'$ and thus from D is (T_0, σ')-maximal follows D is (T, σ')-maximal.

Assume there exists $t, t' \in \mathrm{Sub}(T)$ such that $t\sigma' = t'\sigma'$ but $t \neq t'$. Since $T = \mathrm{Sub}(T_0\theta)$ there exists $t_0, t_0' \in \mathrm{Sub}(T_0)$ such that $t_0\theta \neq t_0'\theta$ but $t_0\theta\sigma' = t_0'\theta\sigma'$. From $\sigma' = \theta\sigma'$ we have an existence of $t_0, t_0' \in \mathrm{Sub}(T_0)$ such that $t_0\theta \neq t_0'\theta$ but $t_0\sigma' = t_0'\sigma'$. This contradicts the fact that θ satisfies $\mathcal{U}$.

Finally, from D is $(\mathcal{S},\sigma)$-compliant and $\sigma = \theta\sigma$ we have D is $(\mathcal{S}\theta,\sigma)$-compliant. $\square$

3.3 Milestone sequence

In addition to retrace the deduction steps performed in D we want to track which terms relevant to $\mathcal{S}$ are deduced in T, and in which order.

Definition 3.3 (Milestone sequence)**.** *Let T be a set of terms and σ be a ground substitution. We say that $\vec{T}$ is the (T,σ)-milestone sequence of a derivation $D = (l_i \to r_i)_{1\le i\le m}$ if $\vec{T} = t_1,\dots,t_n$ is a sequence of maximal length in which each t_i is either of the form $\to t$ or of the form $?t$, with $t \in \mathrm{Sub}(T)$ and there exists a strictly increasing function $\alpha : \{1,\dots,n\} \to \{1,\dots,m\}$ such that for every $1 \le i \le n$ we have:*

1. *if $\vec{T}[i] = ?t$ then $D[\alpha(i)] = ?t\sigma$;*
2. *if $\vec{T}[i] = \to t$ then $D[\alpha(i)] = l_i \to t\sigma$ is a standard deduction rule;*

Lemma 8. *Let $\sigma \models \mathcal{S}$, $T \supseteq \mathrm{Sub}(\mathcal{S})$ and σ' be an extension of σ on $\mathrm{Vars}\,(T)$. Let D be (T,σ')-maximal derivation one-to-one localized by T for σ'. Let $\vec{T}$ be a (T,σ')-milestone sequence.* ***Then*** *for any i for any $x \in \mathrm{Vars}\left(\vec{T}[i]\right)$ there exists $j < i$ such that $\vec{T}[j] = \to x$.*

Proof. If $x \in \mathrm{Vars}\left(\vec{T}[i]\right)$ then there exists corresponding deduction $D[j]$ that deduces term $\vec{T}[i]\sigma'$. Then by Lemma 3 there exists $k < j$ such that $D[j]$ deduces by a standard rule $x\sigma'$. From the injectivity of σ follows that x is the only term of $\mathrm{Sub}(T)$ having σ' image equal $x\sigma'$. Thus, by definition of milestone sequence, there exists $m < i$ such that $\vec{T}[m] = \to x$. $\square$

4 Deciding constraint systems

From now we suppose that the **considered subterm deduction system** contains a rule $x_1, x_2 \to f(x_1, x_2)$, where f is a function symbol with arity 2 that does not occur in any other rule.

Theorem 1. *Let σ such that $\sigma \models \mathcal{S}$, T such that $T \supseteq \mathrm{Sub}(\mathcal{S})$ and σ' an extension of σ on $\mathrm{Vars}\,(T)$. Let D be a (T,σ')-maximal derivation one-to-one localized by T for σ' in which no term is deduced twice by a standard rule.*

Then there exists a solution τ of $\mathcal{S}$ of size polynomial in $|\mathrm{Sub}(T)|$.

Proof. First let us define a *replacement* of a term q by term p in t denoted as $t|_{q\hookleftarrow p}$ as follows: $t|_{q\hookleftarrow p}$ is the term is obtained from t by simultaneous replacing all occurrences of q in t by p. For a substitution $\sigma = \{x \mapsto t_x : x \in \mathrm{dom}\,(\sigma)\}$ we define $\sigma|_{q\hookleftarrow p} = \{x \mapsto (t_x|_{q\hookleftarrow p}) : x \in \mathrm{dom}\,(\sigma)\}$

Let $\vec{T}$ be a (T,σ')-milestone sequence for D.

Let $\vec{M} = m_1, \ldots, m_n$ be the maximal increasing sequence such that for any $i = 1, \ldots, n$, $\vec{T}[m_i] = ?t_{m_i}$. We put also $m_0 = 0$ and $m_{n+1} = |\vec{T}| + 1$. Let $\vec{T_i} = \vec{T}[m_i + 1 : m_{i+1} - 1]$.

GOAL. We will prove the existence of a ground substitution τ', set of terms $T' \supseteq T$ and a derivation D' which is $(\mathcal{S}, \tau)$-compliant, (T', τ)-maximal (where $\tau = \tau'|_{\mathrm{Vars}(\mathcal{S})}$ is of a linear size on $\mathrm{Sub}(T)$) and is one-to-one localized by T' with τ' such that its (T, τ')-milestone sequence coincides with $\vec{T}$.

If it is proved, by Lemma 1 we can show that $\tau \models \mathcal{S}$.

BUILD T'. Let X be the set of variables of $\vec{T}$ whose σ'-instance are not derivable from the empty knowledge. By Lemma 8 each variable x of $\mathrm{Vars}\left(\vec{T}\right)$ appears first as $\to x$ in $\vec{T}$. Therefore, we may put $X = \{x_1, \ldots, x_u\} = \mathrm{Vars}\left(\vec{T}\right) \setminus \left\{x :\to x \in \vec{T_0}\right\}$. Let for each $x \in X$, let $\bar{x}$ be a new fresh variable (corresponding to x) and let $\bar{X} = \{\bar{x} : x \in X\}$. Finally, we put $T' = T \cup \bar{X}$.

BUILD τ'. Let τ' be a ground substitution defined as follows:

- for any $x \in X$, $\bar{x}\tau'$ is a nonce n_x and $x\tau' = f(t_{m_i}\tau', n_x)$, where $\to x$ appears first in $\vec{T_i}$ (note that by Lemma 8 for any $y \in \mathrm{Vars}\,(t_{m_i})$, $\to y$ appears first time at position before m_i in $\vec{T}$ and thus τ' is correctly defined);
- for any $y \in \mathrm{Vars}\left(\vec{T_0}\right)$, $y\tau' = n_y$;
- [2]for any $z \in \mathrm{Vars}\,(T) \setminus \mathrm{Vars}\left(\vec{T}\right)$, $z\tau' = a_z$, where a_z is a fresh constant from $\mathcal{A} \setminus \mathcal{C}_{\mathrm{med}}$ not appearing in $\mathrm{Sub}(T)$.

We can see that $x\tau$ is of polynomial size on $|\mathrm{Sub}(\mathcal{S})|$ for any $x \in \mathrm{Vars}\,(T)$. SHOW τ' IS INJECTIVE ON $\mathrm{Sub}(T')$. Suppose the contrary, let $p, q \in \mathrm{Sub}(T')$ be a pair with minimal size of $p\tau'$ and having $p\tau' = q\tau'$, while $p \neq q$. If neither p nor q is a variable, then this contradicts the minimality of $p\tau'$ (we can choose subterms of p and q satisfying the choice criteria). If both are variables, then it is not possible by the construction of τ'. W.l.o.g. let $p \in \mathcal{X}$ and $q \notin \mathcal{X}$. The case where $p\tau'$ is a nonce or another constant is impossible; thus $p\tau' = f(t_{i_j}, n_x)$ and $q = f(u, \bar{x})$ (since by construction for every nonce n_x there exists only one variable $\bar{x}$ such that $\bar{x}\tau' = n_x$ and $n_x \notin \mathrm{Sub}(T')$). But again, by construction (note that $\bar{x}$ was a fresh variable), the only term in $\mathrm{Sub}(T')$ having $\bar{x}$ as a subterm is $\bar{x}$, thus $q \in \mathcal{X}$: contradiction. $\Diamond$

BUILD A REPLACEMENT TO PASS FROM τ' TO σ'. Let δ be the replacement $\delta = |_{\{x\tau' \hookleftarrow x\sigma' \,:\, x \in \mathrm{Vars}(T)\}}$. Then $\tau'\delta = \sigma'$ on $\mathrm{Vars}\,(T)$. Moreover, from the property we have just proven follows that for any $t \in \mathrm{Sub}(T')$, we have $(t\tau')\delta = t(\tau'\delta)$ and for $t \in \mathrm{Sub}(T)$, we have $(t\tau')\delta = t\sigma'$. Note also that $(x\tau')\delta = x\tau'$ for any $x \in \bar{X}$.

BUILD $(\mathcal{S}, \tau)$-COMPLIANT DERIVATION D' LOCALIZED BY T' WITH τ'. Let $D'_0 = \to n_{x_1}, \ldots, \to n_{x_u}$. Let D'_1 be a sequence of rules of length $|\vec{T}|$ such that

[2]We note that in practice $\mathrm{Vars}\,(T) \setminus \mathrm{Vars}\left(\vec{T}\right) = \emptyset$ if we see how T is constructed in Lemma 7.

for any $i \leq |\vec{T}|$:

- if $\vec{T}[i] = ?t$ then $D'_1[i] = ?t\tau'$;
- if $\vec{T}[i] = \rightarrow x$ and $x \in X$ then $D'_1[i] = n_x, t_{m_j}\tau' \rightarrow x\tau'$, where $\rightarrow x$ appears first in $\vec{T}_j$;
- if $\vec{T}[i] = \rightarrow y$ and $y \in \mathrm{Vars}\left(\vec{T}_0\right)$ then $D'_1[i] = \rightarrow y\tau'$;
- if $\vec{T}[i] = \rightarrow t$ and $t \notin \mathcal{X}$ then since D is one-to-one localized by T, there exists $t_1, \ldots, t_k$ such that $?t_j \in \vec{T}[1 : i-1]$ or $\rightarrow t_j \in \vec{T}[1 : i-1]$ for $j = 1, \ldots, k$ and $t_1, \ldots, t_k \rightarrow t$ is a deduction rule. Thus, we put $D'_1[i] = t_1\tau', \ldots, t_k\tau' \rightarrow t\tau'$.

We define $D' = D'_0, D'_1$. Note that $\mathrm{R}_{D'_0}(|D'_0|) = \bar{X}\tau'$ and for any i, $\mathrm{R}_{D'_1}(i) = \vec{T}[1 : i]\tau'$. Thus, by the construction D' is a derivation which is $(\mathcal{S}, \tau)$-compliant and localized by T' for τ'. Moreover, it is one-to-one localized since τ' is injective on $\mathrm{Sub}(T')$.

We have by construction of D' that its (T', τ')-milestone sequence is $\vec{T}' = \rightarrow \bar{x}_1, \ldots, \rightarrow \bar{x}_u, \vec{T}$. Moreover, $|D'| = |\vec{T}'|$.

Show that D' is (T', τ')-maximal. That is, for any $t \in \mathrm{Sub}(T')$ if $t\tau' \in \mathrm{Der}(\mathrm{R}_{D'}(i))$ then $t\tau' \in \mathrm{R}_{D'}(\mathrm{Next}_{D'}(i) - 1)$.

The case $t \in \bar{X}$ is trivial, since $\bar{X}\tau'$ is deduced at the very beginning of D'.

Suppose that there an exists index j and term $t \in \mathrm{Sub}(T)$ such that $t\tau' \in \mathrm{Der}(t_{m_1}\tau', \ldots, t_{m_j}\tau')$ but $t\tau' \notin \mathrm{R}_{D'}(u + m_{j+1} - 1)$, i.e. $t\tau'$ is not deduced before the next to j non-standard rule in D'. In this case, $t\sigma' \notin \mathrm{Der}(t_{m_1}\sigma', \ldots, t_{m_j}\sigma')$, otherwise by maximality $t\sigma'$ would be deduced before $(j+1)$-th non-standard rule of D and by construction, $t\tau'$ would also appear in D' before $(j+1)$-th nonstandard rule of D'.

Let j be such a minimal index. Note that $\mathrm{Vars}(t) \subseteq \mathrm{Vars}\left(\vec{T}\right)$, otherwise by construction $t\tau'$ would contain some fresh constants from $\mathcal{A} \setminus \mathcal{C}_{\mathrm{med}}$ and thus would not be derivable from $t_{i_1}\tau', \ldots, t_{i_j}\tau'$. Let m' (resp. m) be the maximal index such that $D'[1 : m']$ (resp. $D[1 : m]$) contains exactly j non-standard rules. Thus, $t\tau' \in \mathrm{Der}(\mathrm{R}_{D'}(m'))$ and $t\sigma' \notin \mathrm{Der}(\mathrm{R}_D(m))$. Note that $t\tau' \notin \mathrm{R}_{D'}(m')$ (otherwise it would imply $t\sigma' \in \mathrm{R}_D(m)$). Let E' be a minimal sequence of standard rules such that $D'[1 : m'], E'$ is a derivation ending with a standard deduction of $t\tau'$. W.l.o.g., we suppose that $E'[1 : |E'|-1]$ does not deduce terms from $\mathrm{Sub}(T)\tau'$ (otherwise, if $t'\tau'$ is deduced in $E'[1 : |E'|-1]$ with $t' \in \mathrm{Sub}(T)$ then *(i)* either $t'\sigma' \in \mathrm{Der}(\mathrm{R}_D(m))$ and by maximality of D $t'\sigma' \in \mathrm{R}_D(m)$ which contradicts the minimality of E' *(ii)* or $t'\sigma' \notin \mathrm{Der}(\mathrm{R}_D(m))$ which implies $t'\sigma' \notin \mathrm{R}_D(m)$; thus by construction $t'\tau' \notin \mathrm{R}_{D'}(m')$ and we could chose t' instead of t).

Let $\mathcal{S}'$ be a constraint system obtained from $\mathcal{S}$ by removing all constraints after j-th !-constraint and removing all $\natural$-constraints. By construction, $D'[1 : m'], E'$ is a proof of $\tau \models \mathcal{S}'$ and thus we can apply Lemma 3, i.e. all rules of $E'[1 : |E'|-1]$ are compositions.

Suppose that t is a variable. Note that $t\tau'$ is not a nonce, otherwise by definition of τ', $t \in \vec{T}_0$ and thus $t\sigma' \in \mathrm{R}_D(m)$. Therefore, $t\tau' = f(t_{m_k}\tau', n_t)$, where

$\to t$ first appears in $\vec{T}_k$. Since t is a variable, the last rule of E' is also a composition, more precisely $t_{m_k}\tau', n_t \to f(t_{m_k}\tau', n_t)$. If $k \leq j$, by construction of τ', $t\sigma'$ must be in $\mathrm{R}_D\,(m)$. Thus, $k > j$. Since $D'[1:m'], E'$ is a derivation, either $t_{m_k}\tau' \in \mathrm{R}_{E'}\,(|E'|-1)$ or $t_{m_k}\tau' \in \mathrm{R}_{D'}\,(m')$. The former contradicts the choice of E'. The latter case implies $t_{m_k}\sigma' \in \mathrm{R}_D\,(m) \subseteq \mathrm{Der}(t_{m_1}\sigma', \ldots, t_{m_j}\sigma')$ and thus, as $j < k$ we have that $\mathrm{Der}(t_{m_1}\sigma', \ldots, t_{m_k}\sigma') = \mathrm{Der}(t_{m_1}\sigma', \ldots, t_{m_{k-1}}\sigma')$. Thus, $\vec{T}_k$ must be empty, otherwise it contradicts the maximality of D and that no term is deduced twice by a standard rule in D. This contradicts that $\to t$ appears first in T_k.

Thus, $t \notin \mathcal{X}$.

Let us build a sequence of rules E such that $E[i] = E'[i]\delta$ and show that $D[1:m], D'_0, E$ is a proof of $t\sigma' \in \mathrm{Der}(t_{m_1}\sigma', \ldots, t_{m_j}\sigma')$.

Let us show that $E'[i]\delta$ is a rule.

- If $E'[i] = \to o$ is a nonce generation, then $o \notin \mathrm{img}\,(\tau')$ due to the minimality of E' and since all variables of T' that are mapped to nonces by τ' are deduced in D' before the first non-standard rule. Thus $o\delta = o$ and we have $E[i] = \to o$
- If $E'[i]$ is another composition, then $E'[i] = t'_1, \ldots, t'_v \to h(t'_1, \ldots, t'_v)$. Since $t \notin \mathcal{X}$ and $E'[1:|E'|-1]$ does not deduce terms from $\mathrm{Sub}(T)\tau'$ we have $h(t'_1, \ldots, t'_v) \neq x\tau'$ for any $x \in \mathrm{Vars}\,(T')$. Thus, $h(t'_1, \ldots, t'_v)\delta = h(t'_1\delta, \ldots, t'_v\delta)$ and we have $t'_1\delta, \ldots, t'_v\delta \to h(t'_1\delta, \ldots, t'_v\delta)$ is a composition rule.
- If $E'[i]$ is a decomposition, then since no decomposition rule contains f, the value of $x\tau'$ (which is a fresh nonce or has f as a root symbol) may be replaced with any other term and we still obtain an instance of the same decomposition rule, i.e. $E'[i]\delta$ is an instance of a decomposition rule.

As noted above, since $\forall r \in \mathrm{Sub}(T), (r\tau')\delta = r\sigma'$ and $D'_0\delta = D'_0$ by construction we have $\mathrm{R}_{D'}\,(m')\,\delta \subseteq \mathrm{R}_D\,(m) \cup \left\{x\tau' : x \in \bar{X}\right\}$. Thus, $D[1:m], D'_0, E$ is a derivation deducing $t\tau'\delta = t\sigma'$, i.e. $t\sigma' \in \mathrm{Der}(t_{i_1}\sigma', \ldots, t_{i_j}\sigma')$. Contradiction.

Therefore, D' is (T', τ')-maximal.

Conclusion. Since $\mathrm{Sub}(\mathcal{S}) \subseteq T'$, and τ' is injective on $\mathrm{Sub}(T')$, we have that by construction of D', for any term $t \in \mathrm{Sub}(\mathcal{S})$, $t\sigma'$ is deduced before j-th non-standard rule of D (resp. deduced in D) if and only if $t\tau'$ is deduced before j-th non-standard rule of D' (resp. deduced in D'). Therefore, since $\sigma \models \mathcal{S}$ and D is $(\mathcal{S}, \sigma)$-compliant and $(\mathrm{Sub}(\mathcal{S}), \sigma)$-maximal and since D' is $(\mathcal{S}, \tau)$-compliant and $(\mathrm{Sub}(\mathcal{S}), \tau)$-maximal we may use twice Lemma 1 and obtain that τ satisfies $\mathcal{S}$. □

Corollary 1. *Let $\mathcal{S}$ be a constraint system. $\mathcal{S}$ is satisfiable, if and only if there exists a solution σ' of $\mathcal{S}$ with polynomial size w.r.t.* $|\mathrm{Sub}(\mathcal{S})|$.

Proof. ($\Leftarrow$) is trivial, since $\sigma' \models \mathcal{S}$. Consider ($\Rightarrow$). Let $\sigma \models \mathcal{S}$. By Lemma 7 there exists a set of terms T, a substitution θ both with the size linear in $|\mathrm{Sub}(\mathcal{S})|$

and an extension γ of σ and (T, γ)-maximal $(\mathcal{S}\theta, \sigma)$-compliant derivation D one-to-one localized by T for γ. We also have $\gamma = \theta\gamma$ (which implies $\sigma = \theta\sigma$). Thus σ satisfies $\mathcal{S}\theta$.

From the same lemma we have $\mathrm{Sub}(\mathcal{S}\theta) \subseteq T$. By Theorem 1 there exists a substitution τ of size polynomial in $|\mathrm{Sub}(T)|$ (and consequently, polynomial in $|\mathrm{Sub}(\mathcal{S})|$) such that $\tau \models \mathcal{S}\theta$. From this we have $\theta\tau \models \mathcal{S}$. Moreover, since both θ and τ are of polynomial size on $|\mathrm{Sub}(\mathcal{S})|$, $\sigma' = \theta\tau$ is also of polynomial size on $|\mathrm{Sub}(\mathcal{S})|$ and $\sigma' \models \mathcal{S}$. $\square$

From the previous result we can directly derive an NP decision procedure for constraint systems satisfiability: guess a substitution of polynomial size in $|\mathrm{Sub}(\mathcal{S})|$ and check whether it satisfies $\mathcal{S}$ in polynomial time (see e.g. [1]).

5 Conclusion

We have obtained the first decision procedure for deducibility constraints with negation and we have applied it to the synthesis of mediators subject to non-disclosure policies. It has been implemented as an extension of CL-AtSe [21] for the Dolev-Yao deduction system. On the Loan Origination case study, the prototype generates directly the expected orchestration. Without negative constraints undesired solutions in which the mediator impersonates the clerks were found. More details, including problem specifications, can be found at `http://cassis.loria.fr/Cl-Atse`. As in [1, 5] our definition of subterm deduction systems can be extended to allow ground terms in right-hand sides of decomposition rules even when they are not subterms of left-hand sides and the decidability result remains valid with minor adaptation of the proof. A more challenging extension would be to consider general constraints (as in [3]) with negation.

References

[1] M. Abadi and V. Cortier. Deciding knowledge in security protocols under equational theories. *Theoretical Computer Science*, 367(1–2):2 – 32, 2006.

[2] T. Avanesov, Y. Chevalier, M.A. Mekki, and M. Rusinowitch. Web services verification and prudent implementation. In *DPM/SETOP 2011*, volume 7122 of *Lecture Notes in Computer Science*, pages 173–189. Springer, 2012.

[3] Tigran Avanesov, Yannick Chevalier, Michaël Rusinowitch, and Mathieu Turuani. Satisfiability of general intruder constraints with and without a set constructor. *CoRR*, abs/1103.0220, 2011.

[4] Automated Validation of Trust and Security of Service-Oriented Architectures, AVANTSSAR project. `http://www.avantssar.eu`.

[5] M. Baudet. Deciding security of protocols against off-line guessing attacks. In *Proceedings of CCS'05 conference*, pages 16–25. ACM, 2005.

[6] Y. Chevalier, M.A. Mekki, and M. Rusinowitch. Automatic composition of services with security policies. In *Proceedings of SERVICES I 2008*, SERVICES '08, page 529–537, Washington, DC, USA, 2008. IEEE.

[7] R. Corin, S. Etalle, and A. Saptawijaya. A logic for constraint-based security protocol analysis. In *IEEE Symposium on Security and Privacy (S&P), Berkeley, California, USA*, pages 155–168. IEEE Computer Society, 21-24 May 2006.

[8] G. Costa, P. Degano, and F. Martinelli. Secure service orchestration in open networks. *Journal of Systems Architecture - Embedded Systems Design*, 57(3):231–239, 2011.

[9] D. Dolev and A. Yao. On the security of public key protocols. *Information Theory, IEEE Transactions on*, 29(2):198–208, 1983.

[10] A. Armando et al. The AVANTSSAR platform for the automated validation of trust and security of service-oriented architectures. In *TACAS*, volume 7214 of *Lecture Notes in Computer Science*, pages 267–282. Springer, 2012.

[11] S. Frau and M. Torabi Dashti. Integrated specification and verification of security protocols and policies. In *24th IEEE Computer Security Foundations Symposium, CSF 2011, Cernay-la-Ville, France, 27-29 June*, pages 18–32, 2011.

[12] A. Herzig, E. Lorini, Jomi F. Hübner, and L. Vercouter. A logic of trust and reputation. *Logic Journal of IGPL*, 18(1):214–244, 2010.

[13] M. Kourjieh. *Logical Analysis and Verification of Cryptographic Protocols*. Thèse de doctorat, Université Paul Sabatier, Toulouse, France, décembre 2009.

[14] D. Kähler, R. Küsters, and T. Truderung. Infinite state amc-model checking for cryptographic protocols. *Logic in Computer Science, Symposium on*, 0:181–192, 2007.

[15] E. Lorini and R. Demolombe. Trust and norms in the context of computer security: A logical formalization. In Ron van der Meyden and Leendert van der Torre, editors, *Deontic Logic in Computer Science*, volume 5076 of *Lecture Notes in Computer Science*, pages 50–64. Springer Berlin / Heidelberg, 2008.

[16] C. Lynch and C. Meadows. On the relative soundness of the free algebra model for public key encryption. In *Proceedings of the 2007 FCS-ARSPA Workshop*, volume 125 of *Electronic Notes in Theoretical Computer Science*, pages 43–54, 2005. Available at: `http://profs.sci.univr.it/~vigano/fcs-arspa07/fcs-arspa07.pdf`.

[17] F. Martinelli. Towards an integrated formal analysis for security and trust. In *7th IFIP WG 6.1 International Conference, FMOODS 2005, Athens, Greece, June 15-17, 2005*, pages 115–130, 2005.

[18] David A. McAllester. Automatic recognition of tractability in inference relations. *Journal of the ACM*, 40:284–303, 1993.

[19] Jonathan Millen and Vitaly Shmatikov. Constraint solving for bounded-process cryptographic protocol analysis. In *Proceedings of the 8th ACM conference on Computer and Communications Security*, CCS '01, pages 166–175, New York, NY, USA, 2001. ACM.

[20] Network of Excellence on Engineering Secure Future Internet Software Services and Systems, NESSoS project. `http://www.nessos-project.eu`.

[21] Mathieu Turuani. The CL-Atse Protocol Analyser. In *Term Rewriting and Applications (RTA)*, LNCS 4098, pages 277–286, 2006.

[22] Y.Chevalier, M.A. Mekki, and M. Rusinowitch. Orchestration under security constraints. In *Proceedings of FMCO 2010*, volume 6957 of *Lecture Notes in Computer Science*, pages 23–44. Springer, 2012.

RESEARCH CENTRE
NANCY – GRAND EST

615 rue du Jardin Botanique
CS20101
54603 Villers-lès-Nancy Cedex

Publisher
Inria
Domaine de Voluceau - Rocquencourt
BP 105 - 78153 Le Chesnay Cedex
inria.fr

ISSN 0249-6399